Anwendungssicherheit

Lerneinheit 5: Die OWASP Top Ten 2021

Prof. Dr. Christoph Karg

Studiengang Informatik
Hochschule Aalen

Sommersemester 2025

11.5.2025

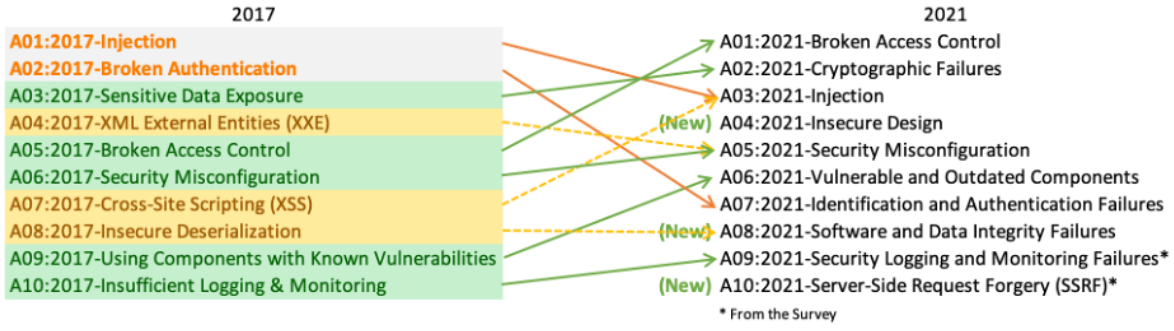
Einleitung

- Das Open Worldwide Application Security Project (OWASP) ist eine Non-Profit-Organisation, die sich der Sicherheit von Software widmet.
- Hinter OWASP steht eine weltweite Community von Firmen, Bildungseinrichtungen und Einzelpersonen.
- Die Arbeit von OWASP umfasst zahlreiche Projekte rund um das Thema Anwendungssicherheit.
- Projekte mit hohem Reifegrad werden als „Flagship Projects“ ausgewiesen.
- Ein Flagship Project ist die OWASP Top Ten, in der die häufigsten und gefährlichsten Schwachstellen in Webanwendungen gelistet werden.
- Für weitere Informationen siehe: `https://owasp.org`

Kategorien der OWASP Top Ten 2021

- A01:2021 – Broken Access Control
- A02:2021 – Cryptographic Failures
- A03:2021 – Injection
- A04:2021 – Insecure Design
- A05:2021 – Security Misconfiguration
- A06:2021 – Vulnerable and Outdated Components
- A07:2021 – Identification and Authentication Failures
- A08:2021 – Software and Data Integrity Failures
- A09:2021 – Security Logging and Monitoring Failures
- A10:2021 – Server-Side Request Forgery (SSRF)

Änderungen zur OWASP Top Ten 2017



Methodik

- Der Ansatz zur Ermittlung der zehn Kategorien ist stark, aber nicht ausschließlich datengestützt.
- Acht Kategorien wurden auf Basis von statistischen Daten ausgewählt.
- Zwei Kategorien (A6 und A10) stammen aus einer Umfrage, die bei Sicherheitsexperten durchgeführt wurde.
- Die National Vulnerability Database und das dort eingesetzten Common Vulnerability Scoring System (CVSS) spielen bei der Bewertung eine wichtige Rolle.
- Für jede Kategorie werden zahlreiche Querbezüge zur Common Weakness Enumeration (CWE) hergestellt.

Common Weakness Enumeration (CWE)

- Die Common Weakness Enumeration (CWE) ist eine Sammlung von Schwachstellen (engl. weakness) von Hard- und Software.
- Für jede dieser Schwachstellen lässt sich ein Schadenspotential (vulnerability) ableiten.
- Ziel: Schulung von Entwicklern und Hardware-Designern zur frühzeitigen Vermeidung der Schwachstellen.
- Für weitere Details siehe: `https://cwe.mitre.org`

Datenfaktoren

- CWEs Mapped ⇝ Anzahl der CWE-Einträge, die der Kategorie zugeordnet wurden
- Incident Rate ⇝ Anteil der Anwendungen in Prozent, die durch die CWEs verwundbar waren
- Avg Weighted Exploit ⇝ Durchschnittlicher Exploit-Subscore der CVSSv2/CVSSv3 der CVEs, die den CWEs zugeordnet wurden
- Avg Weighted Impact ⇝ Durchschnittlicher Impact-Subscore der CVSSv2/CVSSv3 der CVEs, die den CWEs zugeordnet wurden

Datenfaktoren (Forts.)

- Coverage ⇝ Anteil der Anwendungen, die von den teilnehmenden Organisationen bezüglich eines CWE-Eintrags untersucht wurden
- Total Occurrences ⇝ Gesamtzahl der Anwendungen, die von CWEs der Kategorie betroffen waren
- Total CVE ⇝ Gesamtzahl der CVEs in der NVD Datenbank, denen die CWEs der Kategorie zugeordnet waren

Darstellung der Informationen einer Kategorie

- Statistische Daten als Tabelle
- Ursachen für die Schwachstelle
- Tipps zur Vermeidung der Schwachstelle
- Beispiele
- Links zu weiterführenden Informationen
- Verweis auf die zugeordneten CWEs

A01:2021 – Broken Access Control

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
34	**55.97%**	**3.81%**	**6.92**	**5.93**

Max Coverage	Avg Coverage	Total Occurences	Total CVEs
94.55%	**47.72%**	**318.487**	**19.013**

Link: A01:2021

Beschreibung

- Zugriffskontrolle stellt sicher, dass die Benutzer einer Webanwendung nur im Rahmen ihrer zugewiesenen Berechtigungen agieren können.
- Fehlfunktionen führen zu:
  - Unautorisierter Offenlegung vertraulicher Informationen
  - Unautorisierte Änderung von Daten
  - Unautorisiertes Löschen von Daten
  - Unberechtigte Nutzung von Funktionen einer Webanwendung

Nennenswerte CWE-Einträge

- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor ($\rightsquigarrow$ Link)
- CWE-201: Insertion of Sensitive Information Into Sent Data ($\rightsquigarrow$ Link)
- CWE-352: Cross-Site Request Forgery ($\rightsquigarrow$ Link)

Gängige Schwachstellen

- Verletzung des Prinzips der minimalen Rechte (Deny by Default)
- Umgehen von Zugriffskontrolle durch Manipulation der URL, des internen Zustands der Anwendung oder der HTML-Seite
- Zugriff auf den Account eines anderen Nutzers anhand dessen Unique Identifiers (UID)
- Zugriff auf die API ohne die erforderlichen Berechtigungen
- Manipulation von Metadaten
- Fehlkonfiguration von Cross-Origin Resource Sharing (CORS)
- Zugriff eines nicht authentifizierten Nutzers auf eine privilegierte Webseite

Gegenmaßnahmen

- Anwendung der „Deny by Default“ Richtlinie (außer bei öffentlich zugänglichen Informationen)
- Einmalige Implementierung von Mechanismen zur Zugriffskontrolle
- Deaktivieren des Zugriffs auf Verzeichnisse und Metadaten in der Konfiguration des Webservers
- Logging von abgelehnten Zugriffsversuchen und Benachrichtigung der Administratoren
- Begrenzung der Zugriffsrate auf API-Funktionen
- Löschen der Sitzungsinformationen nach dem Ausloggen des Benutzers

A02:2021 – Cryptographic Failures

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
29	**46.44%**	**4.49%**	**7.29**	**6.81**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
79.33%	**34.85%**	**233.788**	**3.075**

Link: A02:2021

Beschreibung

- Diese Kategorie untersucht Schwachstellen, die durch fehlende oder fehlerhaft konfigurierte/implementierte kryptographische Mechanismen entstehen.
- Eine Konsequenz der Ausnutzung von Schwachstellen dieser Art ist die Offenlegung vertraulicher Informationen.
- Es ist zu prüfen, welche Daten während der Übertragung oder der Speicherung abgesichert werden müssen.
- Zu berücksichtigen sind gesetzliche Vorgaben (z.B. DSGVO) oder Regulierungen der Finanzbranche.

Nennenswerte CWE-Einträge

- CWE-259: Use of Hard-coded Password (⇝ Link)
- CWE-327: Broken or Risky Crypto Algorithm (⇝ Link)
- CWE-331: Insufficient Entropy (⇝ Link)

Zu klärende Fragen

- Gibt es Daten, die im Klartext übertragen werden?
- Sind veraltete oder schwache kryptographische Algorithmen im Einsatz?
- Wird einfache Verschlüsselung verwendet, obwohl ein Authenticated-Encryption-Verfahren besser geeignet wäre?
- Sind Default-Passwörter oder schwache Passwörter im Einsatz?
- Wurden Passwörter in ein Repository eingecheckt?
- Werden Serverzertifikate und die zugehörige Trust Chain korrekt validiert?

Zu klärende Fragen (Forts.)

- Ist bei der Verschlüsselung ein unsicherer Betriebsmodus wie beispielsweise ECB im Einsatz?
- Werden Passwörter direkt als kryptographische Schlüssel verwendet, anstatt eine Key Derivation Function einzusetzen?
- Ist der Zufallszahlengenerator für kryptographische Zwecke geeignet und wird er korrekt eingesetzt?
- Sind Fehlermeldungen oder Seitenkanal-Informationen für Angriffe ausnutzbar?

Gegenmaßnahmen

- Identifiziere, welche Daten gemäß gesetzlichen Vorgaben, regulatorischen Vorschriften oder vertraglichen Vereinbarungen zwecks vertraulich sensibel sind.
- Klassifiziere Data-In-Use, Data-At-Rest und Data-In-Transit.
- Speichere sensible Daten nur dann, wenn dies erforderlich ist (Datensparsamkeit).
- Stelle sicher, dass sensible Daten ausschließlich verschlüsselt gespeichert werden.
- Setze ausschließlich aktuelle und als sicher eingestufte kryptographische Verfahren ein.

Gegenmaßnahmen (Forts.)

- Speichere Passwörter unter Einsatz geeigneter Techniken ab.
- Nutze immer Authenticated-Encryption-Verfahren anstatt „normaler“ Verschlüsselung.
- Schlüssel sollten unter Einsatz eines für kryptographische Zwecke geeigneten Zufallszahlengenerators erzeugt werden und im Speicher als Byte-Arrays abgelegt werden.
- Deaktiviere Caching-Mechanismen für sensible Daten.
- Sichere zu übertragende Daten mit geeigneten Verfahren (z.B. TLS) ab.

A03:2021 – Injection

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
33	**19.09%**	**3.37%**	**7.25**	**7.15**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
94.04%	**47.90%**	**274.228**	**32.078**

Link: A03:2021

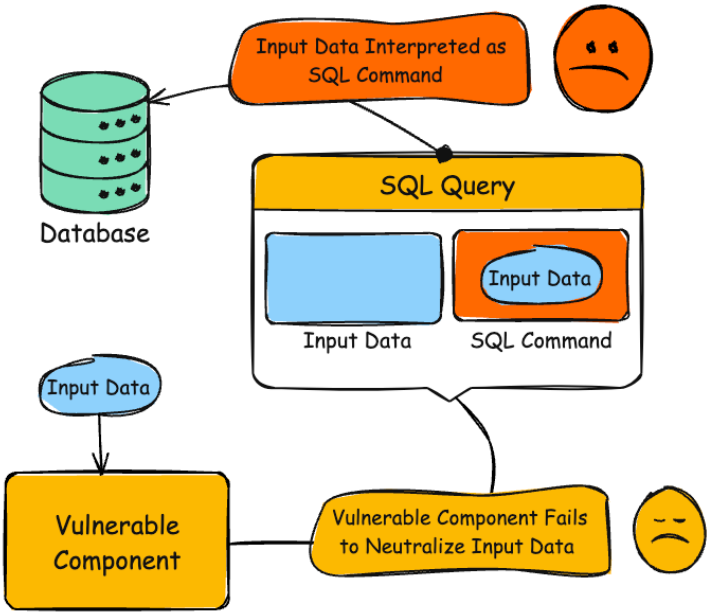
Beschreibung

- Beispiele für diese Art von Schwachstellen sind SQL, NoSQL, OS Command, Object Relational Mapping (ORM) und LDAP Injections.
- Eine Anwendung ist verwundbar, wenn Benutzereingaben nicht überprüft, gefiltert oder bereinigt werden.
- Die beste Methode, um Injection-Schwachstellen zu erkennen, sind Source Code Reviews.
- Automatisierte Tests wie z.B. Fuzzing sind empfehlenswert.

Nennenswerte CWE-Einträge

- CWE-79: Cross-site Scripting ($\rightsquigarrow$ Link)
- CWE-89: SQL Injection ($\rightsquigarrow$ Link)
- CWE-73: External Control of File Name or Path ($\rightsquigarrow$ Link)

Veranschaulichung SQL-Injection (CWE-89)



Quelle: [CWE-89]

Gegenmaßnahmen

- Um Injection Schwachstellen zu verhindern, ist die Trennung von Daten und Befehlen bzw. Anfragen erforderlich.
- Empfehlenswert ist der Einsatz einer sicheren API, die ein passendes Interface bereitstellt und den direkten Zugriff auf den Interpreter verhindert.
- Auf dem Server sollte eine positive Eingabevalidierung eingesetzt werden, z.B. unter Verwendung von regulären Ausdrücken.
- Bei SQL-Anfragen kann `LIMIT` zur Begrenzung der ausgegebenen Datensätze genutzt werden.
- Weitere Empfehlungen findet man in den entsprechenden Cheat-Sheets der OWASP.

A04:2021 – Insecure Design

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
40	**24.19%**	**3.00%**	**6.46**	**6.78**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
77.25%	**42.51%**	**262.407**	**2.691**

Link: A04:2021

Beschreibung

- Diese Kategorie adressiert Risiken, die durch Mängel im Design oder durch Fehler in der Architektur der Software entstehen.
- Sie zeigt den Bedarf an der Nutzung von Threat Modelling, Secure Design Patterns und Referenzarchitekturen auf.
- Unsicheres Design ist von unsicherer Implementierung zu unterscheiden.
- Per Definition kann ein unsicheres Design nicht durch eine perfekte Implementierung repariert werden.
- Eine der Ursachen für unsicheres Design ist die mangelnde Berücksichtigung/Analyse von betriebswirtschaftlichen Risiken.

Sicheres Design

- Sicheres Design steht für eine Methodik, die
  - fortlaufend Bedrohungen analysiert und
  - sicherstellt, dass der entwickelte Code robust gegen aktuelle Angriffstechniken ist.
- Threat Modeling sollte ein fester Bestandteil des Entwicklungsprozesses sein.
- Änderungen im Datenfluss, in der Zugriffskontrolle und anderen Sicherheitskontrollen müssen fortlaufend beobachtet werden.
- Sicheres Design ist nicht als Add-On zur Software-Entwicklung zu verstehen.

Nennenswerte CWE-Einträge

- CWE-209: Generation of Error Message Containing Sensitive Information (⇝ Link)
- CWE-256: Unprotected Storage of Credentials (⇝ Link)
- CWE-501: Trust Boundary Violation (⇝ Link)
- CWE-522: Insufficiently Protected Credentials (⇝ Link)

Gegenmaßnahmen

- Einrichtung und Nutzung eines Secure Development Lifecycles unter Einbindung von AppSec Experten
- Einrichtung und Nutzung von Secure Design Patterns
- Einsatz von Threat Modeling für Authentifizierung, Zugriffskontrolle, Geschäftslogik und Datenflüsse
- Erstellung von Unit Tests zur Prüfung, ob alle kritischen Prozesse resistent gegenüber dem Threat Model sind
- Isolierung der Tiers und Netzwerkschichten gemäß den Sicherheitsanforderungen
- Isolierung der Mandanten durch ein robustes Design
- Begrenzung der Systemressourcen für Nutzer und Dienste

A05:2021 – Security Misconfiguration

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
20	**19.84%**	**4.51%**	**8.12**	**6.56**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
89.58%	**44.84%**	**208.387**	**789**

Link: A05:2021

Beschreibung

- Durch zunehmende Konfigurationsmöglichkeiten in der eingesetzten Software steigt das Risiko von Fehlkonfigurationen.
- Von den getesteten Anwendungen enthielten 4% verschiedenste Arten von Konfigurationsfehlern.
- Beispiele:
  - In einer Produktivumgebung werden bei einem Anwendungsserver die Beispielanwendungen nicht entfernt.
  - Bei einem Webserver wird die Darstellung der Verzeichnisse der Webanwendung nicht deaktiviert.
  - Der Anwendungsserver liefert detaillierte Fehlermeldungen inklusive Stack Traces aus.

Nennenswerte CWE-Einträge

- CWE-16: Configuration ($\rightsquigarrow$ Link)
- CWE-260: Password in Configuration File ($\rightsquigarrow$ Link)
- CWE-526: Cleartext Storage of Sensitive Information in an Environment Variable ($\rightsquigarrow$ Link)
- CWE-611: Improper Restriction of XML External Entity Reference ($\rightsquigarrow$ Link)

Ursachen für Verwundbarkeiten

- Die Systeme sind unzureichend bezüglich Sicherheit gehärtet.
- Auf den Systemen sind nicht benötigte Komponenten installiert bzw. aktiviert.
- Default-Zugänge wurden nicht deaktiviert bzw. die zugehörigen Passwörter wurden nicht geändert.
- Das Error Handling liefert zu detaillierte Fehlermeldungen oder Stack Traces an die Nutzer aus.
- Beim Upgrade eines Systems sind aktuelle Sicherheitsmechanismen nicht aktiviert bzw. falsch konfiguriert.
- Die eingesetzte Software ist veraltet oder enthält bekannte Schwachstellen.

Gegenmaßnahmen

- Ein wiederholbarer Prozess für die Systemhärtung ermöglicht das Aufsetzen neuer System mit entsprechender Absicherung.
- Eine minimale Plattform ohne nicht benötigte Software-Pakete, Dokumentationen und Beispielanwendungen bildet die Grundlage für den Betrieb der Webanwendung.
- Eine Checkliste zur Aktualisierung und Überprüfung der Systeme sollte für alle Komponenten erstellt werden.
- Die Anwendung sollte segmentiert werden, um eine sichere Isolierung der eingesetzten Komponenten und die Mehrmandantenfähigkeit zu erreichen.

A06:2021 – Vulnerable and Outdated Components

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
3	**27.96%**	**8.77%**	**5.0**	**5.0**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
51.78%	**22.47%**	**30.457**	**0**

Link: A06:2021

Beschreibung

- Der Einsatz verwundbarer Software-Komponenten ist eine bekannte Ursache für Schwachstellen in Webanwendungen.
- Die Risikobewertung dieser Tatsache ist je nach Komponente mühsam.

Nennenswerte CWE-Einträge

- CWE-1104: Use of Unmaintained Third-Party Components (⇝ Link)

Ursachen für Verwundbarkeiten

- Die Versionsnummern der eingesetzten Komponenten sowohl auf Client- als auch Server-Seite sind nicht bekannt.
- Die eingesetzte Software ist verwundbar, veraltet oder wird vom Hersteller nicht mehr unterstützt.
- Es werden keine oder nur unregelmäßig Schwachstellen-Scans durchgeführt.
- Es werden keine Informationen über aktuelle Schwachstellen recherchiert.
- Updates und Sicherheitspatches werden nicht zeitnah eingespielt.
- Die Kompatibilität der eingesetzten Komponenten wird nicht oder nur unzureichend getestet.

Gegenmaßnahmen

- Entferne nicht benötigte Software, Dokumentation, Komponenten, Beispielanwendungen, usw.
- Erstelle ein Inventar aller eingesetzten Komponenten (Frameworks, Bibliotheken) sowohl auf Client- als auch auf Server-Seite.
- Beziehe Software ausschließlich von den offiziellen Quellen.
- Überprüfe, ob die genutzten Komponenten noch gepflegt werden und Sicherheits-Patches erhalten.

A07:2021 – Identification and Authentication Failures

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
22	**14.84%**	**2.55%**	**7.40**	**6.50**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
79.51%	**45.72%**	**132.195**	**3.897**

Link: A07:2021

Beschreibung

- Diese Kategorie ist aus der Kategorie „Broken Authentication“ der Top Ten 2017 entstanden.
- Der Kontext ist nun breiter gestaltet und betrachtet zusätzlich auch Fehler in der Identifizierung.

Nennenswerte CWE-Einträge

- CWE-297: Improper Validation of Certificate with Host Mismatch (⇝ Link)
- CWE-287: Improper Authentication (⇝ Link)
- CWE-384: Session Fixation (⇝ Link)

Ursachen für Schwachstellen

- Die Anwendung erlaubt automatisierte Angriffe wie z.B. Brute Force Attacken oder Credential Stuffing.
- Die Anwendung erlaubt die Nutzung schwacher Passwörter.
- Es kommen schwache oder ineffiziente Methoden zur Wiederherstellung des Accounts zum Einsatz.
- Das Verfahren zur Speicherung der Passwörter erfüllt nicht die gängigen Standards.
- Es kommt keine oder wirkungslose Mehr-Faktor-Authentisierung zum Einsatz.
- Die URL enthält Session-Informationen, die für einen Angriff nutzbar sind.
- Session-IDs werden nach erfolgreichem Logout oder einer Inaktivitätsphase nicht korrekt gelöscht.

Gegenmaßnahmen

- Falls möglich, sollte Mehr-Faktor-Authentisierung genutzt werden.
- Auf die Installation von Default-Zugängen und/oder Default-Passwörter sollte verzichtet werden.
- Es sollten Verfahren zur Erkennung schwacher Passwörter eingesetzt werden.
- Die Anforderung an die Qualität von Passwörtern sollte sich an gängigen Empfehlungen orientieren.
- Bei mehreren fehlgeschlagenen Login-Versuchen sollte der nächste Login-Versuch verzögert werden.
- Serverseitig sollte ein Session-Manager eingesetzt werden, der gängige Anforderungen erfüllt.

A08:2021 – Software and Data Integrity Failures

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
10	**16.67%**	**2.05%**	**6.94**	**7.94**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
75.04%	**45.35%**	**47.972**	**1.152**

Link: A08:2021

Beschreibung

- Diese Kategorie in in der OWASP Top Ten neu hinzugekommen.
- Der Fokus liegt auf getroffenen Annahmen im Zusammenhang mit Software Updates, wichtigen Daten und CI/CD Pipelines ohne die entsprechende Datenintegrität zu prüfen.
- Beispiele:
  - In der Anwendung wird Software auf nicht-vertrauenswürdigen Quellen eingesetzt.
  - Objekte oder Daten werden in einer Struktur kodiert oder serialisiert, die ein Angreifer einsehen oder modifizieren kann (Insecure Deserialization).

Nennenswerte CWE-Einträge

- CWE-829: Inclusion of Functionality from Untrusted Control Sphere (⇝ Link)
- CWE-494: Download of Code Without Integrity Check (⇝ Link)
- CWE-502: Deserialization of Untrusted Data (⇝ Link)

Gegenmaßnahmen

- Einsatz von digitalen Signaturen oder ähnlichen Verfahren zur Überprüfung des Ursprungs von Daten oder Software
- Installation von Software ausschließlich über vertrauenswürdige Quellen
- Einrichtung eines Prozesses zur Überprüfung von Änderungen am Code oder der Konfiguration der Anwendung
- Überprüfung, ob die CI/CD-Pipeline eine fehlerfreie Aufgabentrennung, Konfiguration und Zugriffskontrolle besitzt
- Implementierung von Datenserialisierung, die gängige Sicherheitsanforderungen erfüllt

A09:2021 – Security Logging and Monitoring Failures

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
4	**19.23%**	**6.51%**	**6.87**	**4.99**

Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
53.67%	**39.97%**	**53.615**	**242**

Link: A09:2021

Beschreibung

- Die Analyse von Logging oder Monitoring Informationen kann herausfordernd sein.
- Oft ist menschliche Expertise für die Bewertung erforderlich.
- Diese Kategorie wurde aufgrund einer Community-Umfrage in die Top Ten 2021 aufgenommen.
- Für diese Kategorie gibt es kaum CVE/CVSS-Daten.

Nennenswerte CWE-Einträge

- CWE-778: Insufficient Logging (⇝ Link)
- CWE-117: Improper Output Neutralization for Logs (⇝ Link)
- CWE-223: Omission of Security-relevant Information (⇝ Link)
- CWE-532: Insertion of Sensitive Information into Log File (⇝ Link)

Ursachen

- Wichtige Events wie z.B. Logins oder fehlgeschlagene Logins werden nicht protokolliert.
- Warnungen und Fehlermeldungen von Software-Komponenten erzeugen keine oder unverständliche Log-Einträge.
- Log-Daten werden nicht auf auffällige Einträge überprüft.
- Log-Daten werden nur lokal auf dem Server gespeichert.
- Es gibt keine oder nur ineffektive Prozesse zur Behandlung von Fehlermeldungen.
- Automatisierte Penetration-Test-Tools liefern keine Alarme.
- Die Anwendung kann Angriffe nicht zeitnah erkennen.

Gegenmaßnahmen

- Alle Fehler bei Logins, Zugriffen und serverseitiger Eingabe-Validierung müssen protokolliert werden.
- Die Log-Daten sollten in einem Format gespeichert werden, welches von gängiger Analyse-Software verarbeitbar ist.
- Log-Daten müssen auf eine Art und Weise enkodiert werden, dass Angriffe auf die Analyse-Software verhindert werden.
- Es muss sichergestellt sein, dass Finanz-Transaktionen ab einer bestimmten Höhe fälschungssicher protokolliert werden.
- Es wird eine Einrichtung eines Incident Response And Recovery Plans empfohlen.

A10:2021 – Server-Side Request Forgery (SSRF)

CWEs Mapped	Max Incident Rate	Avg Incident Rate	Avg Weighted Exploit	Avg Weighted Impact
1	**2.72%**	**2.72%**	**8.28**	**6.72**

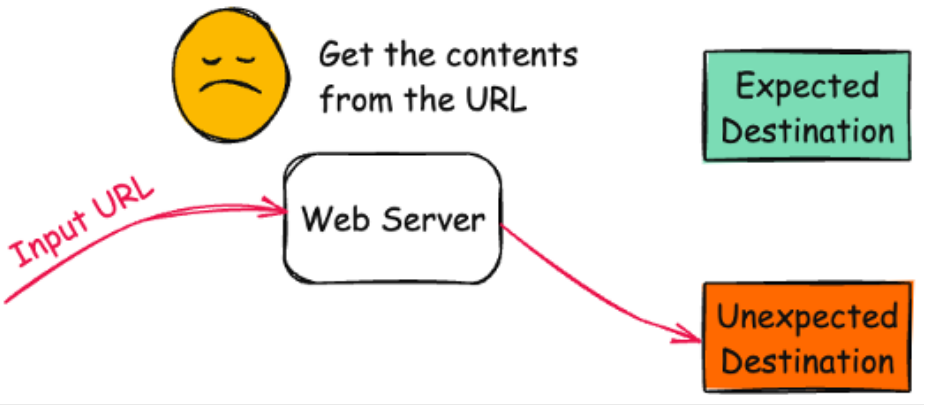
Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
67.72%	**67.72%**	**9.503**	**385**

Link: A10:2021

Beschreibung

- SSRF-Schwachstellen entstehen, wenn ein Server Daten von einer vom Nutzer übergebene URL abruft, ohne diese vorher zu überprüfen.
- Konsequenz: Angreifer können über eine präparierte URL schädliche Daten in die Webanwendung einschleusen.
- Diese Kategorie wurde aus der Community-Umfrage übernommen. In dieser war sie als Nummer 1 gelistet.
- Für diese Kategorie gibt es kaum statistische Informationen und nur wenige CWE-Einträge.

Veranschaulichung SSRF (CWE-918)



Quelle: [CWE-918]

Nennenswerte CWE-Einträge

- CWE-918: Server-Side Request Forgery (SSRF) (⇝ Link)

Gegenmaßnahmen auf Netzwerkebene

- Der Zugriff auf Remote-Ressourcen sollte in einem separaten Netzwerksegment erfolgen, um den Schaden eines erfolgreichen Angriffs zu beschränken.
- Durch das Erzwingen der „Deny By Default“ Regel auf den Firewalls wird nur der freigegebene Datenverkehr zugelassen.
- Firewall-Regeln sollten pro Anwendung einem Eigentümer zugeordnet werden und einen Lebenszyklus besitzen.
- Alle erlaubten und blockierten Datenflüsse müssen protokolliert werden.

Gegenmaßnahmen auf Anwendungsebene

- Alle von einem Client empfangenen Daten müssen überprüft und bereinigt werden.
- Vor der Verarbeitung müssen URLs überprüft werden, z.B. unter Einsatz einer Allow-Liste.
- Die an Clients gesendeten Daten müssen vor dem Versand überprüft werden.
- HTTP-Redirects müssen deaktiviert werden.
- Um Angriffe wie DNS-Rebinding oder „Time Of Check, Time Of Use“ (TOCTOU) zu verhindern, muss die inhaltliche Korrektheit von URLs überprüft werden.

Zusammenfassung

- Die OWASP Top Ten 2021 ist eine Liste der häufigsten in Webanwendungen zu findenden Schwachstellen.
- Viele der Schwachstellen treten auf, obwohl sie einfach zu verhindern wären.
- Neben den in der Top Ten 2021 gelisteten Schwachstellen gibt es noch diverse andere.
- Schwachstellen gibt es nicht nur in Webanwendungen, sondern auch in anderen Komponenten von IT-Systemen.